The X-38 Spacecraft Fault-Tolerant Avionics System

Coy Kouba[1], Deborah Buscher[1], Joseph Busa[2]

1. NASA-Johnson Space Center, Houston, TX
2. Charles Stark Draper Laboratories, Cambridge, MA

ABSTRACT

In 1995 NASA began an experimental program to develop a reusable crew return vehicle (CRV) for the International Space Station. The purpose of the CRV was threefold: (i) to bring home an injured or ill crewmember; (ii) to bring home the entire crew if the Shuttle fleet was grounded; and (iii) to evacuate the crew in the case of an imminent Station threat (i.e., fire, decompression, etc). Built at the Johnson Space Center, were two approach and landing prototypes and one spacecraft demonstrator (called V201). A series of increasingly complex ground subsystem tests were completed, and eight successful high-altitude drop tests were achieved to prove the design concept. In this program, an unprecedented amount of commercial-off-the-shelf technology was utilized in this first crewed spacecraft NASA has built since the Shuttle program. Unfortunately, in 2002 the program was canceled due to changing Agency priorities. The vehicle was 80% complete and the program was shut down in such a manner as to preserve design, development, test and engineering data.

This paper describes the X-38 V201 fault-tolerant avionics system. Based on Draper Laboratory's Byzantine-resilient fault-tolerant parallel processing system and their "network element" hardware, each flight computer exchanges information on a strict timescale to process input data, compare results, and issue voted vehicle output commands. Major accomplishments achieved in this development include: (i) a space qualified two-fault tolerant design using mostly COTS (hardware and operating system); (ii) a single event upset tolerant network element board, (iii) on-the-fly recovery of a failed processor; (iv) use of synched cache; (v) realignment of memory to bring back a failed channel; (vi) flight code automatically generated from the master measurement list; and (vii) built in-house by a team of civil servants and support contractors.

This paper will present an overview of the avionics system and the hardware implementation, as well as the system software and vehicle command & telemetry functions. Potential improvements and lessons learned on this program are also discussed.

I. AVIONICS ARCHITECTURE OVERVIEW

The X-38 V201 avionics architecture is a four string, two-fault tolerant avionics system. The central part of the avionics architecture is the four Flight Critical Computer's (FCCs) and the Network Element Fifth Unit (NEFU). Each FCC consists of a Flight Critical Processor (FCP), an Instrumentation Control Processor (ICP), a Network Element (NE) card, two Multiprotocol/RS-422 cards, four Digital Output (DO) cards, an Analog Output (AO) card, and an IRIG-B/Decomm card. A simplified view of the architecture is pictured below.

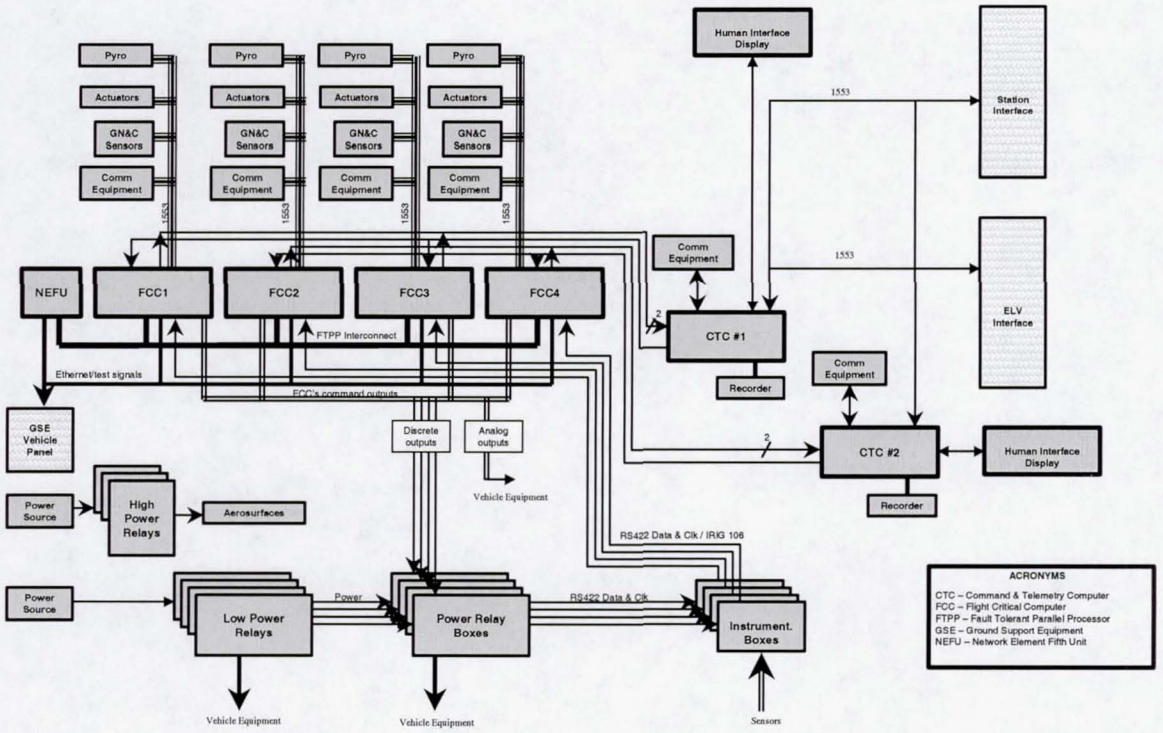


Figure 1: X-38 / V201 Avionics Architecture

The FCP is the main application processor. This board is a Radstone Power PC604R single-board computer that runs the VxWorks operating system. This board contains Draper's Fault Tolerant System Services (FTSS) software, which provides scheduling services, communication services, time services, Fault Detection and Isolation (FDI) services, Redundancy Management (RM) services, and system support services. The FTSS software in combination with the JSC-provided Vehicle, Mission, and Power Management software provides a basic environment in which applications, such as flight control, can execute and meet all necessary timing requirements.

The ICP is the I/O processor. This board is also a Radstone Power PC604R single-board computer that runs the VxWorks operating system. This board obtains the majority of its sensor information from the Data Acquisition Units (DAUs) via an IRIG-B/Decomm

card and the Electromechanical Actuator (EMA) system via a 1553 card. The remainder of the sensor information is obtained via 1553B data buses from the Space Integrated GPS/INS (SIGIs), the Flush Air Data System (FADS), and the S-band Transponder, and via RS-422 from the Altimeters. The ICP also outputs commands to the subsystems. Commands to a few analog devices, such as the cabin fans, are issued via the AO I/F. Commands to many digital devices, such as the power switches, are issued via the DO I/F. EMA position commands are issued via the 1553 interface.

Communication between the FCP and ICP occurs over the NE, through a minimal amount of information shared in the VMEbus shared memory space, and via a syncing interrupt between the FCP and the ICP. The NE is developed by Draper Laboratories of Cambridge, Massachusetts and provides, in combination with the FTSS software, the exchange mechanism for input data and the exchange and voting mechanism for output data.

The NEFU is a fifth computer that contains an ICP and an NE. The NEFU was added to the architecture to provide two-fault tolerance.

The Command and Telemetry Computers (CTCs) serve as the vehicle's primary interfaces to machines and people outside of the vehicle. The two CTC machines interface with the four FCPs via the multi-protocol card's interface RS-422 lines. The CTC machines receive remote commands from several sources, including the ground control center and the Aft Flight Deck Portable Ground Support Computer (PGSC). The CTCs send telemetry data to several destinations, including the ground control center and the Aft Flight Deck PGSC.

II. HARDWARE IMPLEMENTATION

The X-38 flight computers are implemented using the industry standard VME64X protocol. All circuit boards are ruggedized COTS catalog items, with the exception of the Draper network element board. A few components were modified COTS to meet our flight specifications, including the Decom board and the Reed-Solomon board. The chassis enclosures, VME backplanes, and power supply modules were custom designed and built.

The FCC chassis contained an 11-slot VME backplane with two redundant power supply modules. The CTC and NEFU chassis utilized the same five-slot VME backplane with three power supply modules (the CTC had only two power supply modules populated). The internal and external wiring was designed and built in-house, with D38999-series connectors being used on the chassis front panels.

A solid-state 5.1 GB SCSI drive is connected to each CTC to serve as the flight data recorder. It is housed in a removable carrier to facilitate offloading of data after a vehicle

test or mission. Each CTC continuously writes FTPP status data to these drives during normal operation.

A number of FPGAs were utilized on the network element board (and many of the COTS boards too). The NE had three Xilinx and Actel FPGAs to incorporate the state machine design of the NE.

During the development of the X-38, these flight computers underwent a rigorous qualification and acceptance program, including extensive functional testing (using both flight software and special test routines), thermal-vacuum testing, vibration testing, and radiation testing.

III. FAULT-TOLERANCE PARALLEL PROCESSING SYSTEM

FTPP

The FTPP is a fault tolerant parallel processing Byzantine resilient system that is realized by utilizing hardware components known as Network Elements (NE). These NEs act as arbiters that connect a redundant set of computers, each considered fault containment regions, to each other as well as to external systems in a manner that implements Byzantine resilience in a parallel processing environment.

FTSS

The Fault Tolerant System Software (FTSS) is a software layer, woven around the OS and works intimately with the FTPP, which allows the developer of the Flight Application the ability to program as if they were running on one computer. The FTSS handles the low-level ability to run this Application in parallel across separate processors, in lockstep, leaving the developers no concern over this parallel nature or redundancy of the system. The Flight Application simply reads from inputs, which the FTSS ensures is congruent across all computers, then writes their output, which the FTSS votes on and delivers. The developer is further relieved from performing health and monitoring of these systems as FTSS performs intensive fault detection, isolation, and recovery.

Commanding/Telemetry

Commanding and Telemetry were dealt with in a different fashion than the rest of the system. In a perfect world, the telemetry and commanding would have been pumped through the Network Elements via the ICPs. However, telemetry and commanding in combination at high data volume and a 10hz rate would have bogged down the system and could have preempted high priority flight critical data. It was determined to be in the best interest to take these data items off the Network Element path yet adhere to all rules governing the NE. The solution was to pull the commands and send telemetry to/from a separate I/O board called the Multi-Protocol Control Computer (MPCC). Bringing commands into the Flight Critical Computers was accomplished by reading the commands from the MPCC at 10hz from the redundant set, voting on the health of each MPCC, then select two healthiest ones to single source exchange their commands to the NE. It was decided to not vote the telemetry at all through the Network Element.

Theoretically the only difference in the data across the redundant computers was the time-stamping itself of the data. The telemetry was simply pumped out the MPCC board at 10hz to a recording device and eventually transmitted to the ground by a separate system.

Asynchronous I/O in a Parallel Processing Byzantine Resilient Environment
In the X-38 architecture, Telemetry is sent down at 10hz. In general, all I/O performed from the FCP should pass through the ICP via 50hz synchronous pipes and then disseminated appropriately. This however, would create a major bottle-neck to communication services (i.e. Network Element) and would stress the already heavily loaded ICP. Instead, the FCP writes its telemetry to a separate board, known as the MPCC, on the VME back-plane. The MPCC is then commanded to transmit the data stream over RS-422 to a Command and Telemetry Computer (CTC), which in turn transmits this data stream to the ground. Several issues arose in trying to successfully bypass the ICP and thereby levy the load off the Network Element. In a full-up Quad scenario, telemetry would be written to the MPCC across four different back-planes. This, in itself, can be considered four asynchronous events absent the Network Element, especially when moving large amounts of data. Complicating the matter further, an error may occur during communication to the MPCC. This may cause a longer writing duration in a re-send scenario or perhaps complete termination of the write altogether which would cause the process to end early. In all, each FCP may return from the telemetry write at different times and could cause loss of channel synchrony. The Telemetry task residing on the FCP should never cause a loss of channel synchrony, as it is not considered a flight-critical process. The solution implemented was to begin by getting an initial time-stamp from a highly accurate local clock on the FCP that is synchronized with the Quad every 1hz. Then, execute the VME access. Finally, a spin-lock is performed on the clock until a conservative pre-determined maximum timeout value is reached.

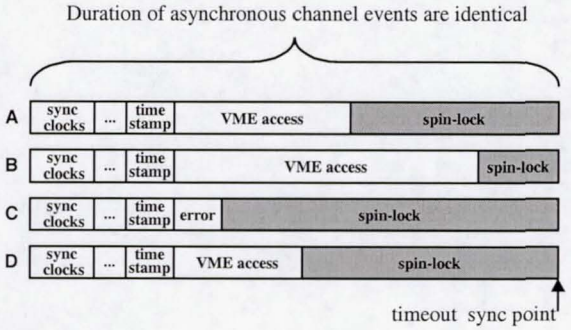


Figure: Asynchronous I/O Events
As shown in the Figure above, in its worst case, this solution allows each of the channels to perform asynchronously, and then finally synchronize when the timeout value is reached.

Software –
The X-38 avionics software consists of five main functions: data acquisition of sensor information, computer processing of sensor information, effector control, telemetry frame

construction, and remote command reception and execution. The FTSS software in combination with the JSC provided Vehicle, Mission, and Power Management software provides a basic environment in which applications, such as flight control, can execute and meet all necessary timing requirements.

The NE interface between the ICP and FCP serves several functions, including the exchange of sensor data from the ICP to the FCP. An example of a two round exchange of a single piece of data from one ICP to four FCPs via the NEs is shown in the figure below. All ICP data is treated as simplex data that is being passed to a quadraplex group. This is due to the fact that the sensors and effectors are not redundant across the four ICPs. Instead, the I/O profile of the X-38 201 vehicle is redundant and/or cross-strapped in only the key areas necessary for vehicle flight, life support, and environmental control.

The figure below shows a single input value being read into the ICP and exchanged via a two round exchange over the NEs to all four FCPs. If that single input value is "bad" (i.e., the sensor has hard-over failed) that "bad" value would be exchanged via the NEs just like any other value. It is up to the application software to determine if the value is "bad."

During the first round of the exchange, the data is sent from one NE to all of the NEs. During the second round of the exchange, the data is sent from all five NEs to all five NEs again. This two round exchange is necessary because 1) the ICPs are not synced during the first exchange (i.e., all four ICPs are running independently and in simplex mode) and 2) the second exchange is necessary to verify that the data exchanged in the first round was received properly by all NEs.

Two Round Exchange Example - On Input Data

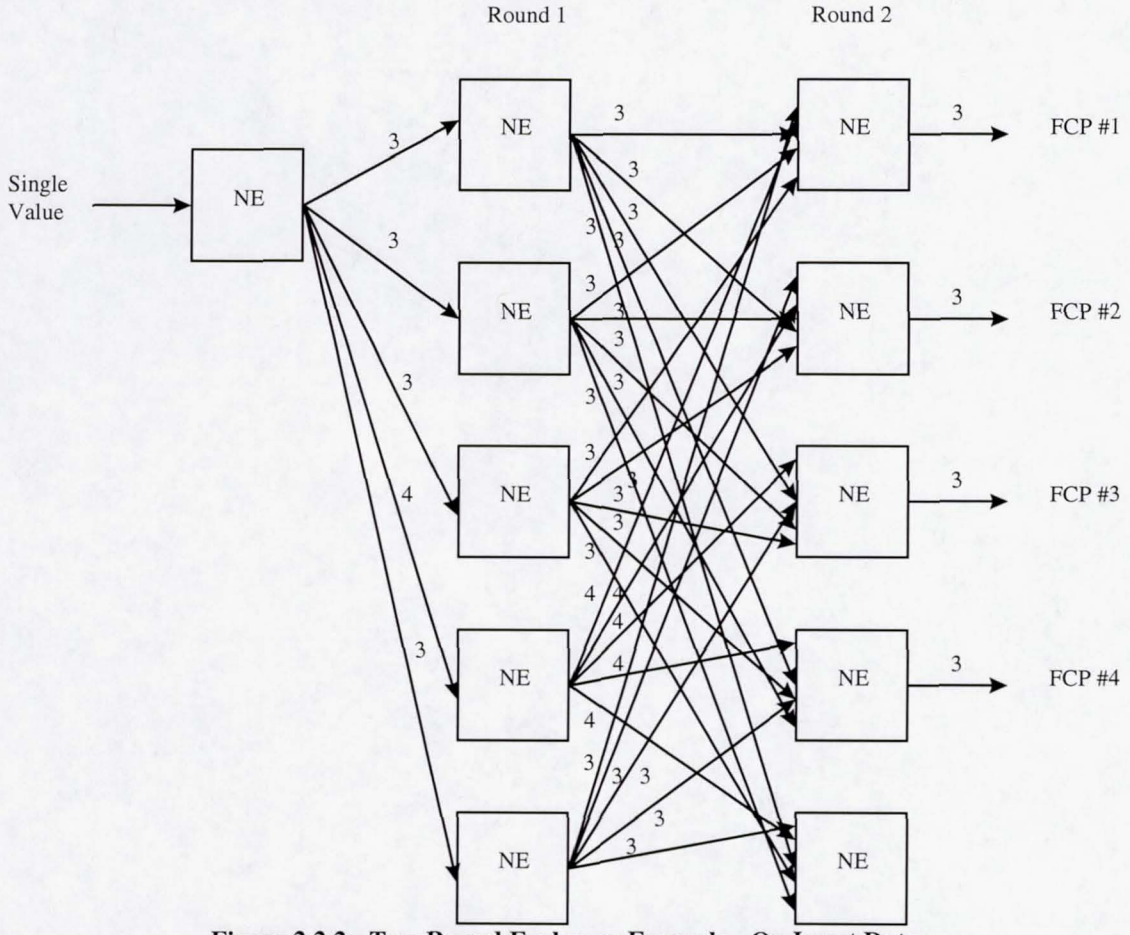


Figure 2.2.2: Two Round Exchange Example - On Input Data

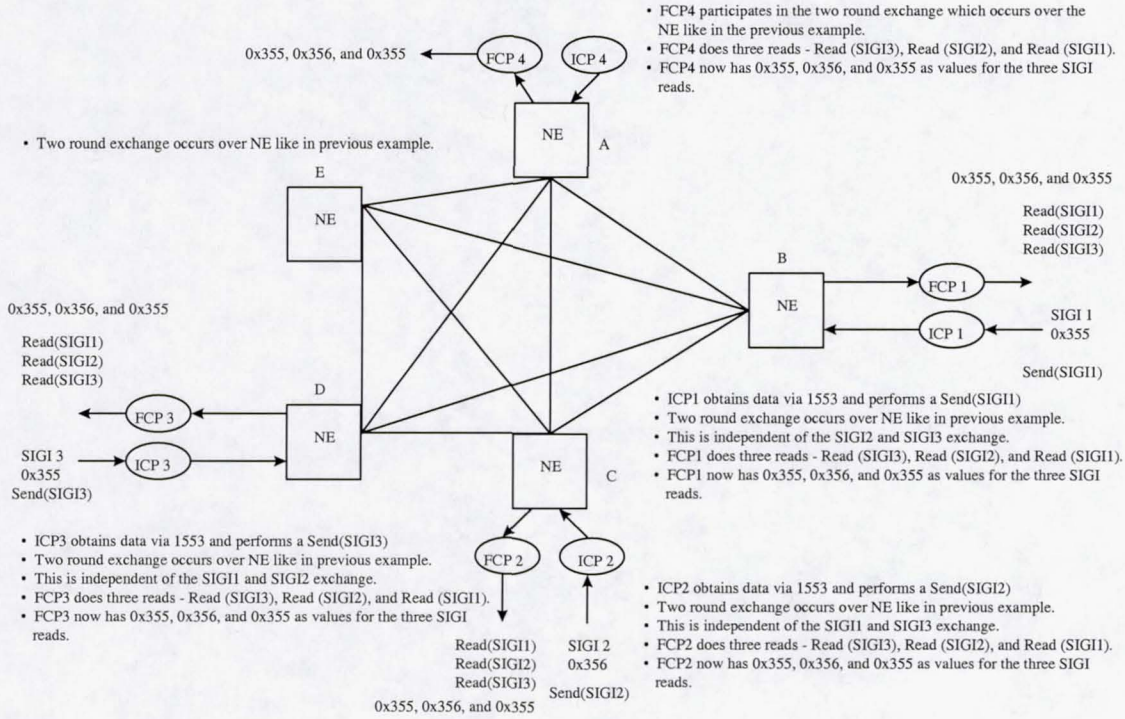
One of the primary jobs of the FCP computer is to run FC and NFC applications. These applications each consist of several parts 1) sensor Subsystem Operating Procedure (SOP) code, which contains sensor data conversion routines, sensor redundancy management routines, and sensor fault detection, isolation, and recovery routines, 2) application code, which takes these sensor inputs and uses them in equations to produce effector commands, 3) effector reverse SOPs, which convert the commands from engineering units to raw effector units, and 4) code for processing remote commands coming from the ground engineers or Shuttle crew.

Each application program is divided up into an initialization procedure, an application code procedure, a sensor SOP procedure, and an effector SOP procedure. Sensor RM and FDIR is included in the sensor SOP task. Each task will operate using a global memory block, which is broken up into 50 Hz, 10 Hz, and 1 Hz data for each subsystem. Only tasks within the same rate group can communicate directly and share data with

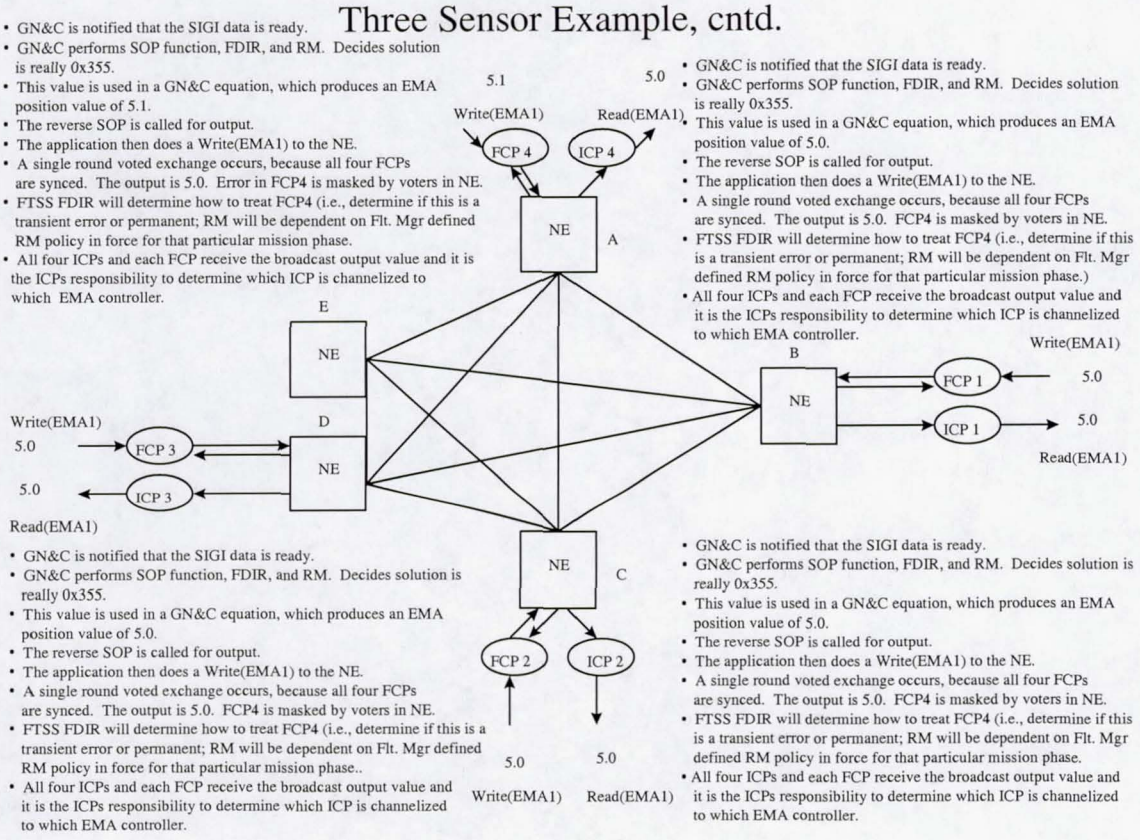
other tasks in that rate group. Data transfer between tasks in a different rate group is performed via FTSS communication services sockets. Since FCP applications do not have access to non-congruent data, FTSS communication services will by-pass the use of the NEs.

The two figures below shows end-to-end how the ICP brings in sensor data, how the application operates on that data and produces an effector command, and how the effector command is output to the ICP.

Three Sensor Example

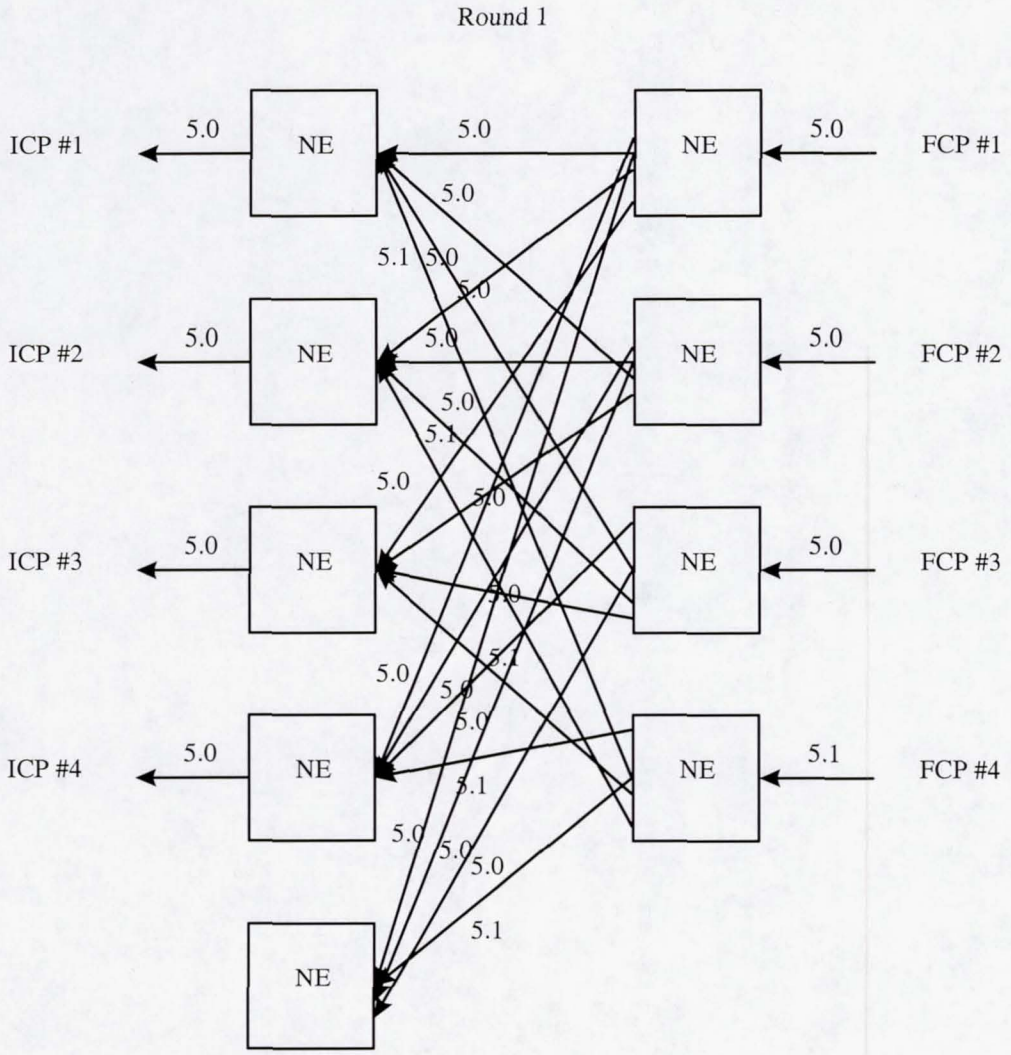


Three Sensor Example, cntd.

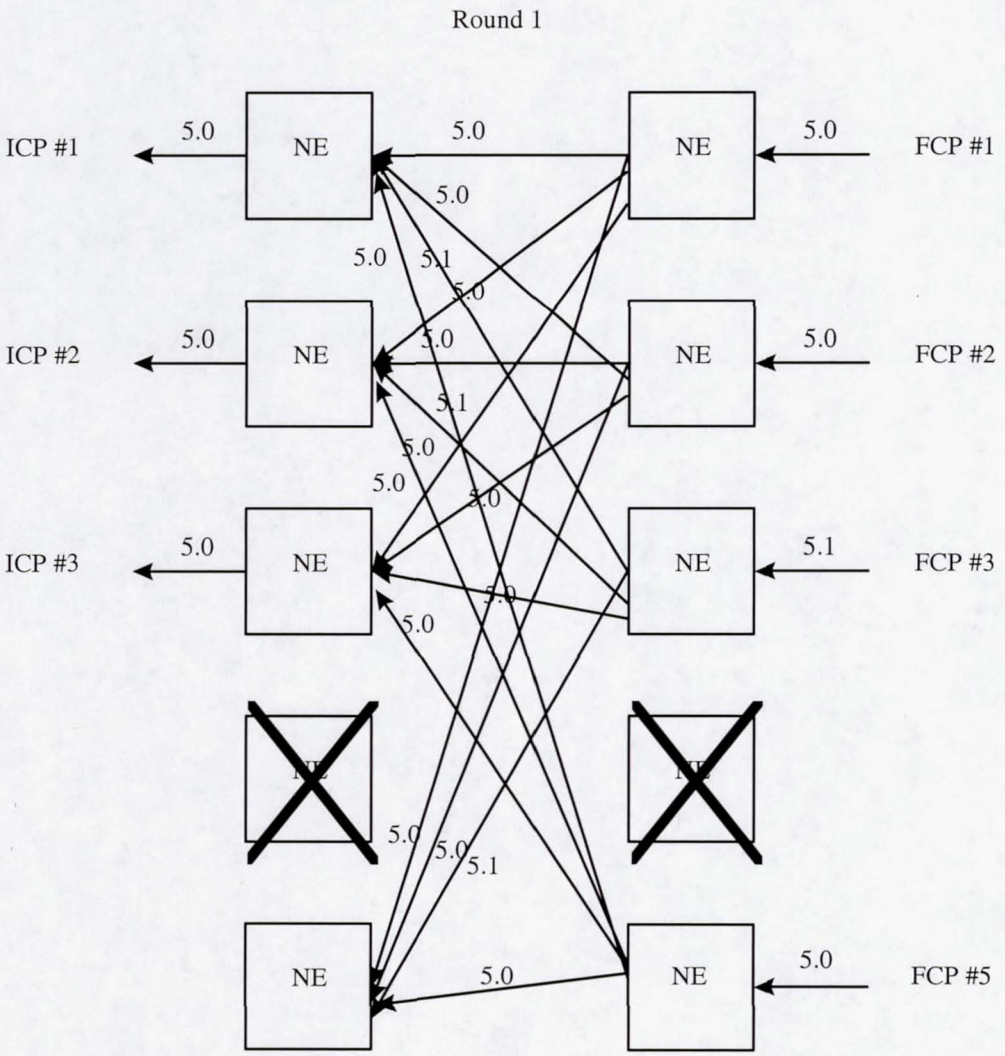


Once the application has completed computation of the sensor data, the application produces an effector command response. The figure below shows how all four FCPs produce the EMA position command at the same time and a single round exchange occurs via the NE. In this case, three of the four FCPs have produced a solution of 5.0. A fourth FCP has produced a solution of 5.1. No FCPs have timed out, so all processors are in sync. Upon the completion of the single round exchange, a voted output is sent (i.e., 5.0) to all four ICPs. The 5.1 position that FCP #4 produced is masked out. The voted output is broadcast to all FCPs and to all four ICPs. This voted broadcast allows both the ICPs to receive the output command and the FCPs to 1) receive the output command, which can then be placed in the telemetry stream, and 2) receive any syndrome data on the output vote, which will in turn be used in FTSS FDI to determine whether or not a processor or NE has a problem and needs to be voted out or powered off. All of the ICPs receive all commands. This allows the FCP, for the most part, to be independent from the effector configuration. It is the ICP's responsibility to know their own identity and what I/O devices are attached to them.

EMA Position 1 - Output Exchange



EMA Position 1 - Output Exchange



Each FCP communicates with one CTC computer via two multi-protocol card RS-422 lines: one for command reading and one for telemetry writing. The figure below shows the connectivity between the CTCs and the FCPs. Note: The FCP/CTC combination is maximized to minimize the chance of two failures (i.e., two FCCs failed) in two fault containment regions bringing down both CTCs. The channelization of the CTCs is completed along with power fault containment region lines.

There are three telemetry gathering (which are known as the data collector tasks) tasks (50, 10, and 1 Hz tasks) and one data logger task (a 10 Hz task) which send the telemetry frame to the CTC. The 50 Hz data collector task gathers all telemetry information at a 50

Hz rate and passes it to the 10 Hz data logger task. The 10 Hz data collector task gathers all telemetry information at a 10 Hz rate and passes it to the 10 Hz data logger task. The 1 Hz data collector task gathers all telemetry information at a 1 Hz rate and passes it to the 10 Hz data logger task. The 10 Hz data logger task then constructs each telemetry frame and, at a 10 Hz rate, outputs a telemetry stream of data to each CTC.

After the telemetry stream write is complete the FCPs read data from the CTC to which they are attached. There will always be a command (even if it is a null command or a repeated command) and status data available to be read.

IV. FUTURE WORK & CONCLUSIONS

The current configuration in the X-38 is fully Byzantine resilient up to the I/O Processor. After that, due to cost and weight concerns, the flow beyond is susceptible to Byzantine errors; though each hardware instance has a minimum of 1 fault tolerance. Improving this system would include implementing the Byzantine philosophy throughout the entire breadth of the system, beyond the Flight Critical components.

Potential improvements to the hardware could be:

(i) Removing the fiber optic links: these components are very fragile to handling and are damaged easily. The fiber optic components also required us to significantly increase the size of each flight chassis due to the minimum bend radius of each fiber cable. An alternative would be to replace them with copper connects using optocouplers to provide isolation.

(ii) Improve the Network Element's throughput to reduce the overloading bottleneck. This would require a Draper design change.

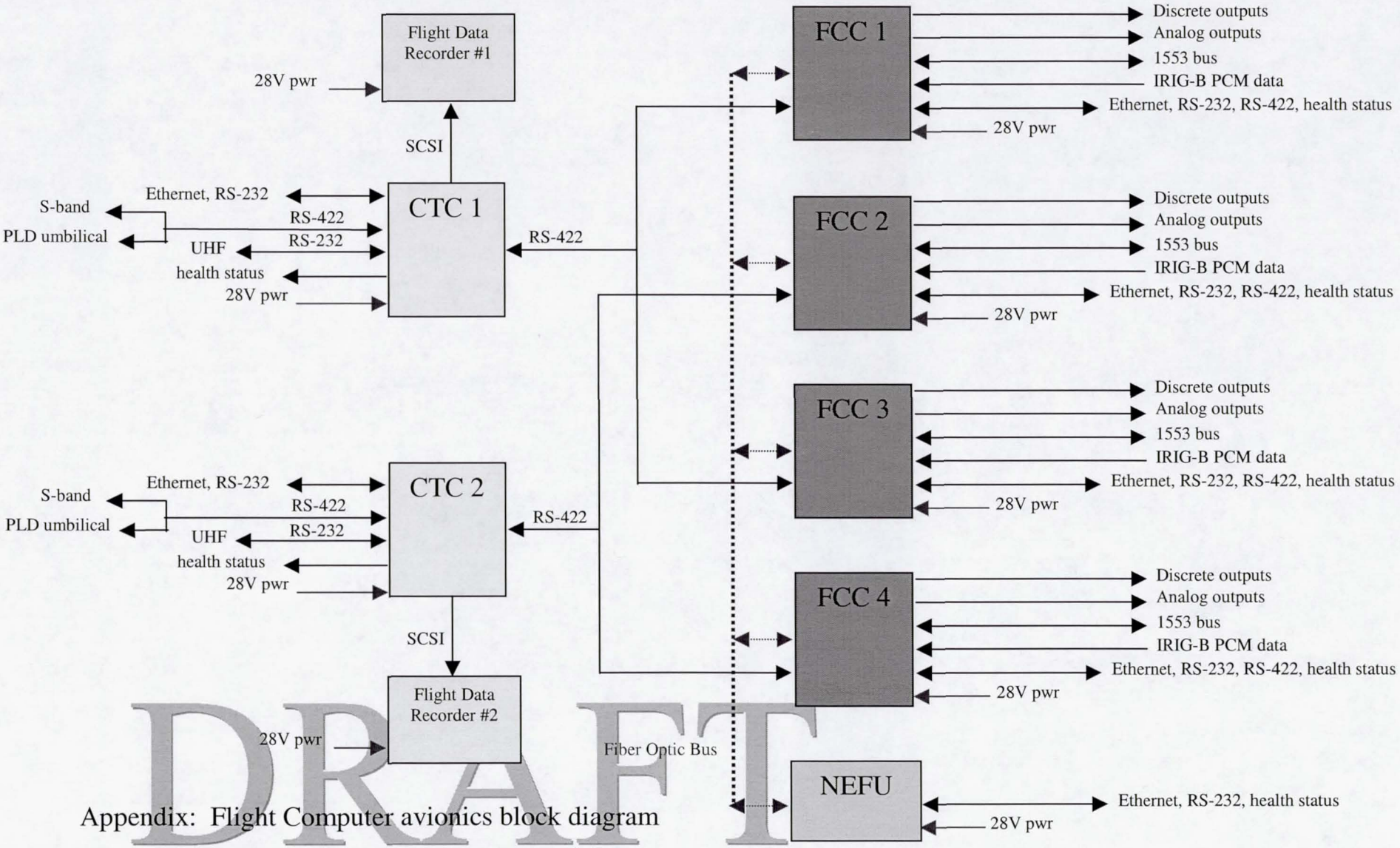
(iii) Use faster FCP and ICP processor boards to also increase throughput

(iv) Implement greater radiation tolerance by upgrading certain parts on the Network Element .

(v) Recent improvements in ruggedized COTS components could also lead to a faster and smaller hardware implementation.

REFERENCES

JSC29309, X-38 Vehicle 201 Software Architecture Definition Document, Version 2.3, December 1999, Buscher, Humphrey, Carvajal



Appendix: Flight Computer avionics block diagram